Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M4
Core Size	32-Bit Single-Core
Speed	120MHz
Connectivity	CANbus, EBI/EMI, Ethernet, IrDA, SD, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	117
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 16x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4e16ea-an

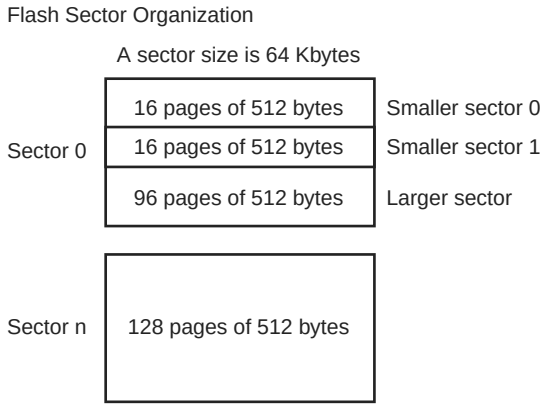
For sector 0:

- The smaller sector 0 has 16 pages of 512 bytes
- The smaller sector 1 has 16 pages of 512 bytes
- The larger sector has 96 pages of 512 bytes

From Sector 1 to n:

The rest of the array is composed of 64 Kbyte sector of each 128 pages of 512 bytes. Refer to Figure 7-3.

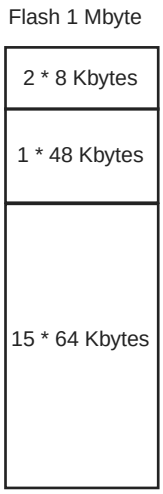
Figure 7-3. Flash Sector Organization



Flash size varies by product. The Flash size of SAM4E device is 1024 Kbytes.

Refer to Figure 7-4 for the organization of the Flash following its size.

Figure 7-4. Flash Size



The following erase commands can be used depending on the sector size:

- 8 Kbyte small sector
  - Erase and write page (EWP)
  - Erase and write page and lock (EWPL)
  - Erase sector (ES) with FARG set to a page number in the sector to erase
  - Erase pages (EPA) with FARG [1:0] = 0 to erase four pages or FARG [1:0] = 1 to erase eight pages. FARG [1:0] = 2 and FARG [1:0] = 3 must not be used.
- 48 Kbyte and 64 Kbyte sectors
  - One block of 8 pages inside any sector, with the command Erase pages (EPA) with FARG[1:0] = 1
  - One block of 16 pages inside any sector, with the command Erase pages (EPA) and FARG[1:0] = 2
  - One block of 32 pages inside any sector, with the command Erase pages (EPA) and FARG[1:0] = 3
  - One sector with the command Erase sector (ES) and FARG set to a page number in the sector to erase
- Entire memory plane
  - The entire Flash, with the command Erase all (EA).

The write commands of the Flash cannot be used under 330 kHz.

7.2.3.2 Enhanced Embedded Flash Controller

The Enhanced Embedded Flash Controller manages accesses performed by the masters of the system. It enables reading the Flash and writing the write buffer. It also contains a User Interface, mapped on the APB.

The Enhanced Embedded Flash Controller ensures the interface of the Flash block.

It manages the programming, erasing, locking and unlocking sequences of the Flash using a full set of commands.

One of the commands returns the embedded Flash descriptor definition that informs the system about the Flash organization, thus making the software generic.

7.2.3.3 Flash Speed

The user needs to set the number of wait states depending on the frequency used:

For more details, refer to the “AC Characteristics” section of the product “Electrical Characteristics”.

Target for the Flash speed at 0 wait state: 24 MHz.

7.2.3.4 Lock Regions

Several lock bits are used to protect write and erase operations on lock regions. A lock region is composed of several consecutive pages, and each lock region has its associated lock bit.

Table 7-1. Lock Bit Number

Product	Number of lock bits	Lock region size
SAM4E	128	8 Kbytes

If a locked-region’s erase or program command occurs, the command is aborted and the EEFC triggers an interrupt.

The lock bits are software programmable through the EEFC User Interface. The command “Set Lock Bit” enables the protection. The command “Clear Lock Bit” unlocks the lock region.

Asserting the ERASE pin clears the lock bits, thus unlocking the entire Flash.

An interrupt request from a peripheral or from software can change the state of the corresponding interrupt to pending.

Active

An exception is being serviced by the processor but has not completed.

An exception handler can interrupt the execution of another exception handler. In this case, both exceptions are in the active state.

Active and Pending

The exception is being serviced by the processor and there is a pending exception from the same source.

11.4.3.2 Exception Types

The exception types are:

Reset

Reset is invoked on power up or a warm reset. The exception model treats reset as a special form of exception. When reset is asserted, the operation of the processor stops, potentially at any point in an instruction. When reset is deasserted, execution restarts from the address provided by the reset entry in the vector table. Execution restarts as privileged execution in Thread mode.

Non Maskable Interrupt (NMI)

A non maskable interrupt (NMI) can be signalled by a peripheral or triggered by software. This is the highest priority exception other than reset. It is permanently enabled and has a fixed priority of -2.

NMIs cannot be:

- Masked or prevented from activation by any other exception.
- Preempted by any exception other than Reset.

Hard Fault

A hard fault is an exception that occurs because of an error during exception processing, or because an exception cannot be managed by any other exception mechanism. Hard Faults have a fixed priority of -1, meaning they have higher priority than any exception with configurable priority.

Memory Management Fault (MemManage)

A Memory Management Fault is an exception that occurs because of a memory protection related fault. The MPU or the fixed memory protection constraints determines this fault, for both instruction and data memory transactions. This fault is used to abort instruction accesses to *Execute Never* (XN) memory regions, even if the MPU is disabled.

Bus Fault

A Bus Fault is an exception that occurs because of a memory related fault for an instruction or data memory transaction. This might be from an error detected on a bus in the memory system.

Usage Fault

A Usage Fault is an exception that occurs because of a fault related to an instruction execution. This includes:

- An undefined instruction
- An illegal unaligned access
- An invalid state on instruction execution
- An error on exception return.

The following can cause a Usage Fault when the core is configured to report them:

- An unaligned address on word and halfword memory access
- A division by zero.

11.6.7.2 SSAT16 and USAT16

Signed Saturate and Unsigned Saturate to any bit position for two halfwords.

Syntax

```
op{cond} Rd, #n, Rm
```

where:

op	is one of:
	SSAT16 Saturates a signed halfword value to a signed range.
	USAT16 Saturates a signed halfword value to an unsigned range.
cond	is an optional condition code, see “Conditional Execution” .
Rd	is the destination register.
n	specifies the bit position to saturate to:
n ranges from 1	n ranges from 0 to 15 for USAT.
to 16 for SSAT	
Rm	is the register containing the value to saturate.

Operation

The SSAT16 instruction:

Saturates two signed 16-bit halfword values of the register with the value to saturate from selected by the bit position in *n*.

Writes the results as two signed 16-bit halfwords to the destination register.

The USAT16 instruction:

Saturates two unsigned 16-bit halfword values of the register with the value to saturate from selected by the bit position in *n*.

Writes the results as two unsigned halfwords in the destination register.

Restrictions

Do not use SP and do not use PC.

Condition Flags

These instructions do not affect the condition code flags.

If saturation occurs, these instructions set the Q flag to 1.

Examples

```
SSAT16    R7, #9, R2   ; Saturates the top and bottom highwords of R2
                       ; as 9-bit values, writes to corresponding halfword
                       ; of R7
USAT16NE  R0, #13, R5  ; Conditionally saturates the top and bottom
                       ; halfwords of R5 as 13-bit values, writes to
                       ; corresponding halfword of R0.
```

11.6.11.8 VFMA, VFMS

Floating-point Fused Multiply Accumulate and Subtract.

Syntax

```
VFMA{cond}.F32 {Sd,} Sn, Sm
VFMS{cond}.F32 {Sd,} Sn, Sm
```

where:

cond is an optional condition code, see “Conditional Execution” .

Sd is the destination register.

Sn, Sm are the operand registers.

Operation

The VFMA instruction:

1. Multiplies the floating-point values in the operand registers.
2. Accumulates the results into the destination register.

The result of the multiply is not rounded before the accumulation.

The VFMS instruction:

1. Negates the first operand register.
2. Multiplies the floating-point values of the first and second operand registers.
3. Adds the products to the destination register.
4. Places the results in the destination register.

The result of the multiply is not rounded before the addition.

Restrictions

There are no restrictions.

Condition Flags

These instructions do not change the flags.

17.5.2 Reinforced Safety Watchdog Timer Mode Register

Name: RSWDT_MR

Address: 0x400E1904

Access: Read-write Once

31	30	29	28	27	26	25	24
–	–	WDIDLEHLT	WDDBGHLT	WDD			
23	22	21	20	19	18	17	16
WDD							
15	14	13	12	11	10	9	8
WDDIS	WDRPROC	WDRSTEN	WDFIEN	WDV			
7	6	5	4	3	2	1	0
WDV							

Note: The first write access prevents any further modification of the value of this register; read accesses remain possible.

Note: The WDD and WDV values must not be modified within three slow clock periods following a restart of the watchdog performed by means of a write access in the RSWDT_CR, else the watchdog may trigger an end of period earlier than expected.

- **WDV: Watchdog Counter Value**

Defines the value loaded in the 12-bit watchdog counter.

- **WDFIEN: Watchdog Fault Interrupt Enable**

0: A Watchdog fault (underflow or error) has no effect on interrupt.

1: A Watchdog fault (underflow or error) asserts interrupt.

- **WDRSTEN: Watchdog Reset Enable**

0: A Watchdog fault (underflow or error) has no effect on the resets.

1: A Watchdog fault (underflow or error) triggers a watchdog reset.

- **WDRPROC: Watchdog Reset Processor**

0: If WDRSTEN is 1, a watchdog fault (underflow or error) activates all resets.

1: If WDRSTEN is 1, a watchdog fault (underflow or error) activates the processor reset.

- **WDD: Watchdog Delta Value**

Defines the permitted range for reloading the RSWDT.

If the RSWDT value is less than or equal to WDD, writing RSWDT_CR with WDRSTT = 1 restarts the timer.

If the RSWDT value is greater than WDD, writing RSWDT_CR with WDRSTT = 1 causes a Watchdog error.

- **WDDBGHLT: Watchdog Debug Halt**

0: The RSWDT runs when the processor is in debug state.

1: The RSWDT stops when the processor is in debug state.

25.8.9 DMAC Error, Buffer Transfer and Chained Buffer Transfer Status Register

Name: DMAC_EBCISR

Address: 0x400C0024

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	ERR3	ERR2	ERR1	ERR0

15	14	13	12	11	10	9	8
–	–	–	–	CBTC3	CBTC2	CBTC1	CBTC0

7	6	5	4	3	2	1	0
–	–	–	–	BTC3	BTC2	BTC1	BTC0

- **BTCx: Buffer Transfer Completed [3:0]**

When BTC[*i*] is set, Channel *i* buffer transfer has terminated.

- **CBTCx: Chained Buffer Transfer Completed [3:0]**

When CBTC[*i*] is set, Channel *i* Chained buffer has terminated. LLI Fetch operation is disabled.

- **ERRx: Access Error [3:0]**

When ERR[*i*] is set, Channel *i* has detected an AHB Read or Write Error Access.

Table 30-2. Authorized Input Data Registers

Operation Mode	Input Data Registers to Write
64-bit CFB	AES_IDATAR0 and AES_IDATAR1
32-bit CFB	AES_IDATAR0
16-bit CFB	AES_IDATAR0
8-bit CFB	AES_IDATAR0
CTR	All

Note: In 64-bit CFB mode, writing to AES_IDATAR2 and AES_IDATAR3 is not allowed and may lead to errors in processing.

Note: In 32, 16, and 8-bit CFB modes, writing to AES_IDATAR1, AES_IDATAR2 and AES_IDATAR3 is not allowed and may lead to errors in processing.

30.4.4.2 Auto Mode

The Auto Mode is similar to the manual one, except that in this mode, as soon as the correct number of AES_IDATARx is written, processing is automatically started without any action in the AES_CR.

30.4.4.3 DMA Mode

The DMA Controller can be used in association with the AES to perform an encryption/decryption of a buffer without any action by software during processing.

The SMOD field in the AES_MR must be configured to 0x2 and the DMA must be configured with non-incremental addresses.

The start address of any transfer descriptor must be configured with the address of AES_IDATAR0.

The DMA chunk size configuration depends on the AES mode of operation and is listed in Table 30-3 “DMA Data Transfer Type for the Different Operation Modes”.

When writing data to AES with a first DMA channel, data are first fetched from a memory buffer (source data). It is recommended to configure the size of source data to “words” even for CFB modes. On the contrary, the destination data size depends on the mode of operation. When reading data from the AES with the second DMA channel, the source data is the data read from AES and data destination is the memory buffer. In this case, the source data size depends on the AES mode of operation and is listed in Table 30-3.

Table 30-3. DMA Data Transfer Type for the Different Operation Modes

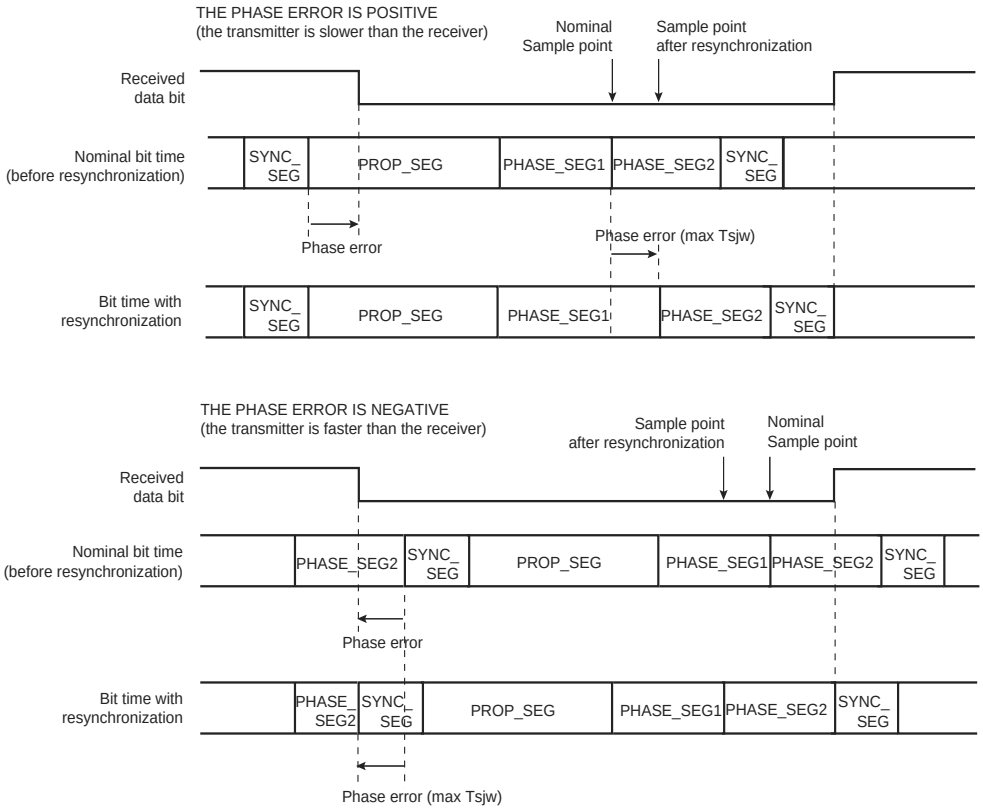
Operation Mode	Chunk Size	Destination/Source Data Transfer Type
ECB	4	Word
CBC	4	Word
OFB	4	Word
CFB 128-bit	4	Word
CFB 64-bit	1	Word
CFB 32-bit	1	Word
CFB 16-bit	1	Half-word
CFB 8-bit	1	Byte
CTR	4	Word

30.4.5 Last Output Data Mode

This mode is used to generate cryptographic checksums on data (MAC) by means of cipher block chaining encryption algorithm (CBC-MAC algorithm for example).

- the phase error is negative, then PHASE_SEG2 is shortened by an amount equal to the resynchronization jump width.

Figure 31-6. CAN Resynchronization



Autobaud Mode

The autobaud feature is enabled by setting the ABM field in the CAN_MR. In this mode, the CAN controller is only listening to the line without acknowledging the received messages. It can not send any message. The errors flags are updated. The bit timing can be adjusted until no error occurs (good configuration found). In this mode, the error counters are frozen. To go back to the standard mode, the ABM bit must be cleared in the CAN_MR.

31.7.4.2 Error Detection

There are five different error types that are not mutually exclusive. Each error concerns only specific fields of the CAN data frame (refer to the Bosch CAN specification for their correspondence):

- CRC error (CERR bit in the CAN_SR): With the CRC, the transmitter calculates a checksum for the CRC bit sequence from the Start of Frame bit until the end of the Data Field. This CRC sequence is transmitted in the CRC field of the Data or Remote Frame.
- Bit-stuffing error (SERR bit in the CAN_SR): If a node detects a sixth consecutive equal bit level during the bit-stuffing area of a frame, it generates an Error Frame starting with the next bit-time.
- Bit error (BERR bit in CAN_SR): A bit error occurs if a transmitter sends a dominant bit but detects a recessive bit on the bus line, or if it sends a recessive bit but detects a dominant bit on the bus line. An error frame is generated and starts with the next bit time.
- Form Error (FERR bit in the CAN_SR): If a transmitter detects a dominant bit in one of the fix-formatted segments CRC Delimiter, ACK Delimiter or End of Frame, a form error has occurred and an error frame is generated.
- Acknowledgment error (AERR bit in the CAN_SR): The transmitter checks the Acknowledge Slot, which is transmitted by the transmitting node as a recessive bit, contains a dominant bit. If this is the case, at least

33.6.14 PIO Interrupt Enable Register

Name: PIO_IER

Address: 0x400E0E40 (PIOA), 0x400E1040 (PIOB), 0x400E1240 (PIOC), 0x400E1440 (PIOD), 0x400E1640 (PIOE)

Access: Write-only

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

- **P0–P31: Input Change Interrupt Enable**

0: No effect.

1: Enables the input change interrupt on the I/O line.

37.7 Universal Synchronous Asynchronous Receiver Transmitter (USART) User Interface

Table 37-15. Register Mapping

Offset	Register	Name	Access	Reset
0x0000	Control Register	US_CR	Write-only	–
0x0004	Mode Register	US_MR	Read/Write	0x0
0x0008	Interrupt Enable Register	US_IER	Write-only	–
0x000C	Interrupt Disable Register	US_IDR	Write-only	–
0x0010	Interrupt Mask Register	US_IMR	Read-only	0x0
0x0014	Channel Status Register	US_CSR	Read-only	0x0
0x0018	Receive Holding Register	US_RHR	Read-only	0x0
0x001C	Transmit Holding Register	US_THR	Write-only	–
0x0020	Baud Rate Generator Register	US_BRGR	Read/Write	0x0
0x0024	Receiver Time-out Register	US_RTOR	Read/Write	0x0
0x0028	Transmitter Timeguard Register	US_TTGR	Read/Write	0x0
0x002C–0x003C	Reserved	–	–	–
0x0040	FI DI Ratio Register	US_FIDI	Read/Write	0x174
0x0044	Number of Errors Register	US_NER	Read-only	0x0
0x0048	Reserved	–	–	–
0x004C	IrDA Filter Register	US_IF	Read/Write	0x0
0x0050	Manchester Configuration Register	US_MAN	Read/Write	0x30011004
0x0054–0x005C	Reserved	–	–	–
0x0060–0x00E0	Reserved	–	–	–
0x00E4	Write Protection Mode Register	US_WPMR	Read/Write	0x0
0x00E8	Write Protection Status Register	US_WPSR	Read-only	0x0
0x00EC–0x00FC	Reserved	–	–	–
0x0100–0x0128	Reserved for PDC Registers	–	–	–

38.7.5 TC Register AB

Name: TC_RABx [x=0..2]

Address: 0x4009000C (0)[0], 0x4009004C (0)[1], 0x4009008C (0)[2], 0x4009400C (1)[0], 0x4009404C (1)[1], 0x4009408C (1)[2], 0x4009800C (2)[0], 0x4009804C (2)[1], 0x4009808C (2)[2]

Access: Read-only

31	30	29	28	27	26	25	24
RAB							

23	22	21	20	19	18	17	16
RAB							

15	14	13	12	11	10	9	8
RAB							

7	6	5	4	3	2	1	0
RAB							

- **RAB: Register A or Register B**

RAB contains the next unread capture Register A or Register B value in real time. It is usually read by the DMA after a request due to a valid load edge on TIOAx.

When DMA is used, the RAB register address must be configured as source address of the transfer.

39.7.10 PWM DMA Register

Name: PWM_DMAR

Access: Write- only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
DMADUTY							

15	14	13	12	11	10	9	8
DMADUTY							

7	6	5	4	3	2	1	0
DMADUTY							

Only the first 16 bits (channel counter size) are significant.

- **DMADUTY: Duty-Cycle Holding Register for DMA Access**

Each write access to PWM_DMAR sequentially updates the CDTY field of PWM_CDTYx with DMADUTY (only for channel configured as synchronous). See “Method 3: Automatic write of duty-cycle values and automatic trigger of the update” .

40.14.11 HSMCI Transmit Data Register

Name: HSMCI_TDR

Address: 0x40080034

Access: Write-only

31	30	29	28	27	26	25	24
DATA							
23	22	21	20	19	18	17	16
DATA							
15	14	13	12	11	10	9	8
DATA							
7	6	5	4	3	2	1	0
DATA							

- **DATA: Data to Write**

Table 46-3. 1.2V Voltage Regulator Characteristics

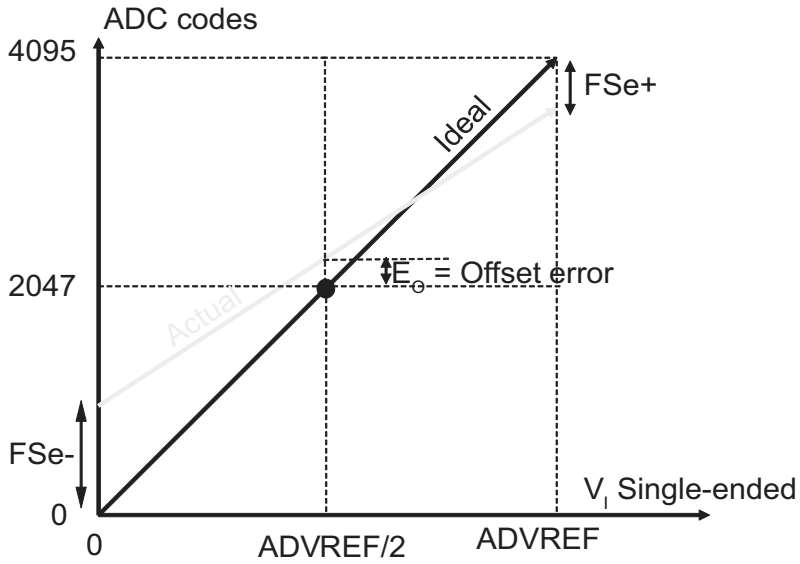
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDIN}	DC Input Voltage Range	(4)	1.6	3.3	3.6	V
V_{DDOUT}	DC Output Voltage	Normal Mode	—	1.2	—	V
		Standby Mode	—	0	—	
$V_{O(accuracy)}$	Output Voltage Accuracy	I_{LOAD} = 0.8 mA to 80 mA (after trimming)	-4	—	4	%
I_{LOAD}	Maximum DC Output Current	V_{DDIN} > 1.8V	—	—	120	mA
		$V_{DDIN} \leq$ 1.8V	—	—	70	
$I_{LOAD-START}$	Maximum Peak Current during startup	(3)	—	—	400	mA
$V_{DROPOUT}$	Dropout Voltage	V_{DDIN} = 1.6V I_{LOAD} = 70 mA	—	400		mV
V_{LINE}	Line Regulation	V_{DDIN} from 2.7 to 3.6 V I_{LOAD} max	—	10	30	mV
$V_{LINE-TR}$	Transient Line Regulation	V_{DDIN} from 2.7 to 3.6 V I_{LOAD} Max $t_r = t_f$ = 5 µs CD_{OUT} = 4.7 µF	—	50	150	mV
V_{LOAD}	Load Regulation	$V_{DDIN} \geq$ 1.8 V I_{LOAD} = 10% to 90% max	—	25	60	mV
$V_{LOAD-TR}$	Transient Load Regulation	V_{DDIN} 1.8 V I_{LOAD} = 10% to 90% max $t_r = t_f$ = 5 µs CD_{OUT} = 4.7 µF	—	45	210	mV
I_Q	Quiescent Current	Normal Mode, @ I_{LOAD} = 0 mA	—	5.5	—	µA
		Normal Mode, @ I_{LOAD} = 120 mA	—	350	—	
		Standby Mode	—	0.06	—	
CD_{IN}	Input Decoupling Capacitor	(1)	—	4.7	—	µF
CD_{OUT}	Output Decoupling Capacitor	(2)	1.85	2.2	5.9	µF
		ESR	0.1	—	10	Ω
t_{on}	Turn on Time	CD_{OUT} = 2.2 µF V_{DDOUT} reaches 1.2V (± 3%)	—	300	—	µs
t_{off}	Turn off Time	CD_{OUT} = 2.2 µF V_{DDIN} 1.8V	—	—	9.5	ms

Notes:
1. A 4.7 µF or higher ceramic capacitor must be connected between VDDIN and the closest GND pin of the device. This large decoupling capacitor is mandatory to reduce startup current, improving transient response and noise rejection.
2. To ensure stability, an external 2.2 µF output capacitor, CD_{OUT} must be connected between the VDDOUT and the closest GND pin of the device. The ESR (Equivalent Series Resistance) of the capacitor must be in the range 0.1Ω to 10Ω. Solid tantalum and multilayer ceramic capacitors are all suitable as output capacitor. A 100 nF bypass capacitor between VDDOUT and the closest GND pin of the device helps decrease output noise and improves the load transient response.
3. Defined as the current needed to charge external bypass/decoupling capacitor network.
4. See Section 5.2.2 “VDDIO Versus VDDIN”

Single-ended Mode

Figure 46-17 illustrates the ADC output code relative to an input voltage V_I between 0V (Ground) and V_{ADVREF}. The ADC is configured in Single-ended mode by connecting internally the negative differential input to $V_{ADVREF}/2$. As the ADC continues to work internally in Differential mode, the offset is measured at $V_{ADVREF}/2$.

Figure 46-17. Gain and Offset Errors in Single-ended Mode



where:

- FSe = (FSe+) - (FSe-) is for full-scale error, unit is LSB code
- Offset error E_O is the offset error measured for V_I = 0V
- Gain error $E_G = 100 \times FSe / 4096$, unit in %

The error values in Table 46-37 and Table 46-38 include the sample and hold error as well as the PGA gain error.

Table 46-37. Single-ended Gain Error

Offset Mode	OFFx = 0		OFFx = 0		OFFx = 1		OFFx = 0		OFFx = 1	
Gain Mode	**1**		**2**		**2**		**4**		**4**	
AutoCorrection	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
Average Gain Error (%)	0.449	0.078	0.771	-0.010	0.781	0.117	1.069	-0.029	1.064	0.151
Standard Deviation (%)	0.420	0.200	0.430	0.313	0.425	0.327	0.420	0.415	0.415	0.371
Min Value (%)	-0.811	-0.522	-0.518	-0.947	-0.493	-0.864	-0.190	-1.274	-0.181	-0.962
Max Value (%)	1.709	0.679	2.061	0.928	2.056	1.099	2.329	1.216	2.310	1.265

Table 46-38. Single-ended Output Offset Error

Offset Mode	OFFx = 0	OFFx = 0	OFFx = 1	OFFx = 0	OFFx = 1
Gain	**1**	**2**	**2**	**4**	**4**
Average Offset Error (LSB)	-5.7	-7.7	-10.3	-7.3	-18.7
Standard Deviation (LSB)	1.8	3.9	3.4	6	7
Min Value (LSB)	-11.1	-19.4	-20.5	-25.3	-39.7
Max Value (LSB)	-0.3	4	-0.1	10.7	2.3

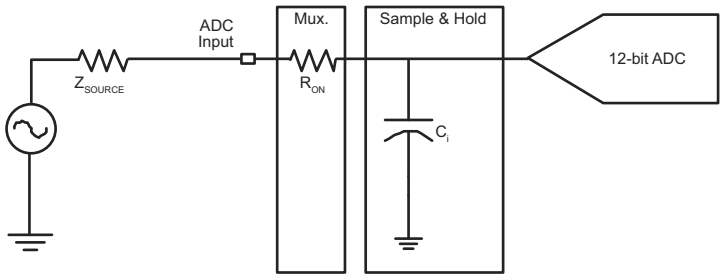
Table 46-44. Z_i Input Impedance

f_S (MHz)	1	0.5	0.25	0.125	0.0625	0.03125	0.015625	0.007813
C_i = 2 pF								
Z_i (MΩ)	0.5	1	2	4	8	16	32	64
C_i = 4 pF								
Z_i (MΩ)	0.25	0.5	1	2	4	8	16	32
C_i = 8 pF								
Z_i (MΩ)	0.125	0.25	0.5	1	2	4	8	16

Track and Hold Time versus Source Output Impedance

Figure 46-19 shows a simplified acquisition path.

Figure 46-19. Simplified Acquisition Path



During the tracking phase, the ADC needs to track the input signal during the tracking time shown below:

$$t_{TRACK} = 0.054 \times Z_{SOURCE} + 205$$

with t_{TRACK} expressed in ns and Z_{SOURCE} expressed in Ω.

The ADC already includes a tracking time of 15 t_{CP_ADC}

Two cases must be considered:

- If the calculated tracking time (t_{TRACK}) is lower than 15 t_{CP_ADC}, then AFEC_MR.TRACKTIM can be set to 0.
- If the calculated tracking time (t_{TRACK}) is higher than 15 t_{CP_ADC}, then AFEC_MR.TRACKTIM must be set to the correct value.

46.7.5.4 AFE DAC Offset Compensation

Table 46-45. AFE DAC Offset Compensation

Parameter	Conditions	Min	Typ	Max	Unit
Resolution - N	—	—	12	—	bits
INL	Range [32 to 4063]	-4	—	+4	LSB
DNL	—	-2	—	+2	LSB
LSB relative to V_{REFIN} Scale	LSB = V_{REFIN} /2e12	—	732	—	µV

Table 51-7. SAM4E Datasheet Rev. 11157B 25-April-2013 Revision History (Continued)

Doc. Date	Changes
25-Apr-2013	PIO: Section 33.5 “Functional Description”, added pull-down resistor and registers in Figure 33-5 “Input Glitch Filter Timing”. Section 33.7.46 “PIO Write Protect Mode Register”, replaced the WPKEY bitfield description with a table. Added missing dashes for reserved registers in Table 33-3 “Register Mapping”. Replaced “DIVx” with “DIV” in Section 33.7.29 “PIO Slow Clock Divider Debouncing Register”. Updated the SCHMITTx bitfield description in Section 33.7.48 “PIO Schmitt Trigger Register” and updated the Delayx bitfield description in Section 33.7.49 “PIO I/O Delay Register”.
	SPI: Section 34.7.3.2 “Master Mode Flow Diagram”, added TDRE references in Figure 34-8 “PDC Status Register Flags Behavior”. Section 34.7.4 “SPI Slave Mode”, updated the next-to-last paragraph (“Then, a new data is loaded...”). Replaced offset 0x4C with 0x40, 0xE8 with 0xEC in Section 34.8 “Serial Peripheral Interface (SPI) User Interface”.
	USART: Added a paragraph on IRDA_FILTER programming criteria in Section 37.7.5.3 “IrDA Demodulator” and in the corresponding bitfield description in Section 37.8.20 “USART IrDA FILTER Register”. Section 37.8.18 “USART FI DI RATIO Register”, expanded FI_DI_RATIO bitfield to 16 bits in the register table. Added RXBUFF and TXBUFE bitfields and their descriptions in: - Section 37.8.6 “USART Interrupt Enable Register (SPI_MODE)” - Section 37.8.8 “USART Interrupt Disable Register (SPI_MODE)” - Section 37.8.10 “USART Interrupt Mask Register (SPI_MODE)” - Section 37.8.12 “USART Channel Status Register (SPI_MODE)”
	TC: Fixed a typo in Section 38.1 “Description”: “TIOA1” --> “TIOB1”.
	ACC: Added Table 42-2 “Analog Comparator Inputs”.